Wester's test suite in MuPAD 1.2.2

Paul Zimmermann

April 10, 1995

A few months ago, Michael Wester made a review [4] of the mathematical capabilities of different computer algebra systems, namely Axiom, Derive, Macsyma, Maple, Mathematica and Reduce. This review, which is available by anonymous ftp from `math.unm.edu`, file `pub/cas/Paper.ps`, consists of 131 tests in different domains of mathematics (arithmetic, algebraic equations, differential equations, integration, operator computation, series expansions, limits).

We describe in this paper the problems that can be solved with MuPAD 1.2.2, and how to solve them. The problems marked as [New] are solved using new functionalities of the version 1.2.2 with respect to 1.2.1.

1 Numbers

Problem 1: compute 50!.

```
>> fact(50);

    30414093201713378043612608166064768844377641568960512000000000000
```

Problem 2: compute the prime decomposition of 50!.

```
>> ifactor(fact(50));

 [1, 2, 47, 3, 22, 5, 12, 7, 8, 11, 4, 13, 3, 17, 2, 19, 2, 23, 2, 29, 1,

                  31, 1, 37, 1, 41, 1, 43, 1, 47, 1]
```

The function `ifactor(n)` returns a list $[\epsilon, p_1, \alpha_1, \ldots, p_k, \alpha_k]$ where $\epsilon = \pm 1$, and $n = \epsilon p_1^{\alpha_1} \ldots p_k^{\alpha_k}$.

Problem 3: compute $1/2 + \cdots + 1/10$.

```
>> _plus(1/i$i=2..10);

                              4861/2520
```

In MuPAD each arithmetic operator has a functional form, here `_plus` for addition, so for example `_plus(a,b)` is the same as `a+b`.

Problem 4: compute a numerical approximation of $e^{\pi\sqrt{163}}$ to 50 digits.

```
>> DIGITS:=50: float(exp(PI*sqrt(163)));

        2.6253741264076874399999999999925007259719818568887e17
```

The floating-point numbers are output in scientific notation. This notation can be used as input too.

[New] **Problem 5:** compute a numerical approximation of $J_2(1+i)$, where J is the Bessel function of the first kind.

```
>> besselJ(2,1.0+I);

                    0.4157988694e-1 + 0.2473976415 I
```

Please note that floating-point numbers are "contagious": no `float` instruction is needed here.

[New] **Problem 6:** compute an infinite decimal representation of $1/7$.

```
>> loadlib("numlib"): export(numlib):
>> decimal(1/7);

                          0, [1, 4, 2, 8, 5, 7]
```

This result means that the decimal expansion of $1/7$ is $0.142857142857142\ldots$. Here the instruction `loadlib("numlib")` loads the package. Then `export(numlib)` exports the functions of this package, i.e. enables one to call them with a short name (otherwise we should write `numlib::decimal`).

[New] **Problem 7:** compute the first terms of the continued fraction of π.

```
>> contfrac(PI,5);

                                   1
                        ---------------------- + 3
                                 1
                        ------------------ + 7
                               1
                        ------------- + 15
                             1
                        --------- + 1
                          1
                        --- + 292
                        ...
```

The second (optional) argument of `contfrac` specifies the number of digits to be taken into account before computing the continued fraction.

[New] **Problem 8:** simplify $\sqrt{2\sqrt{3}+4}$.

```
>> radsimp(sqrt(2*sqrt(3)+4));

                                   1/2
                                  3    + 1
```

[New] **Problem 9:** simplify $\sqrt{14+3\sqrt{3+2\sqrt{5-12\sqrt{3-2\sqrt{2}}}}}$.

```
>> radsimp(sqrt(14 + 3*sqrt(3 + 2*sqrt(5 - 12*sqrt(3 - 2*sqrt(2))))));

                                   1/2
                                  2    + 3
```

[New] **Problem 10:** simplify $2\infty - 3$.

```
>> 2*infinity-3;

                                infinity
```

The symbol `infinity` is implemented as a domain in MuPAD. This allows the overloading of the basic arithmetic operations, together with the comparisons. For example we can write in a MuPAD program `if a<infinity then ... end_if`.

2 Statistics

[New] **Problem 11:** compute the standard deviation of the sample $[1, 2, 3, 4, 5]$.

```
>> loadlib("stats"):
>> stats::stdev([1,2,3,4,5],Sample);
```

```
                          1/2  1/2
                         2    5
                         ---------
                             2
```

Here we have shown the other way to call package functions, without exporting them.

3 Algebra

Problem 14: compute the normal form of $(x^2 - 4)/(x^2 + 4x + 4)$.

```
>> normal((x^2 - 4)/(x^2 + 4*x + 4));
```

```
                           x - 2
                           -----
                           x + 2
```

[New] **Problem 15:** simplify $(e^x - 1)/(e^{x/2} + 1)$.

```
>> simplify((exp(x)-1)/(exp(x/2)+1),exp);
```

```
                              / x \
                          exp | - | - 1
                              \ 2 /
```

Problem 16: expand $(x + 1)^{20}$, then differentiate and factorize.

```
>> Factor(diff(expand((x + 1)^20),x));
```

```
                                      19
                          20 (x + 1)
```

MuPAD provides two factorization functions: the toplevel function `Factor` returns an expression, and the library function `factor` returns a list $[a, p_1, \alpha_1, \ldots, p_k, \alpha_k]$ where a is a scalar, the p_i are irreducible polynomials and the α_i are the corresponding powers. In the above example, `factor` would return `[20, x + 1, 19]`.

Problem 17: factor $x^{100} - 1$.

```
>> Factor(x^100-1);
```

```
                / 2     \ /       2    3    4    \ /          2    3    4
(x + 1) (x - 1) \ x  + 1 / \ x + x  + x  + x  + 1 / \ - x + x  - x  + x  +

   \ /    2    4    6    8      \ /  5    10    15    20     \ /     5
 1  / \ - x  + x  - x  + x  + 1  / \ x  + x   + x   + x   + 1 / \ - x  + x

          10    15    20      \ /    10    20    30    40      \
       - x   + x   + x   + 1  / \ - x   + x   - x   + x   + 1  /
```

Problem 19: factor the polynom $x^4 - 3x^2 + 1$ modulo 5.

```
>> factor(poly(x^4 - 3*x^2 + 1,[x],IntMod(5)));

    [1, poly(x - 2, [x], IntMod(5)), 2, poly(x + 2, [x], IntMod(5)), 2]
```

This result means that $x^4 - 3x^2 + 1$ factors into $(x-2)^2(x+2)^2$ modulo 5. MuPAD provides a special representation `poly` for polynomials, that enables the user to specify the variables and the coefficient field.

[New] **Problem 20:** compute the partial fraction decomposition of $(x^2 + 2x + 3)/(x^3 + 4x^2 + 5x + 2)$.

```
>> partfrac((x^2+2*x+3)/(x^3+4*x^2+5*x+2));

                        2       3                - 2
                    - ----- + ----- + 2 (x + 1)
                      x + 1   x + 2
```

4 Trigonometry

[New] **Problem 25:** rewrite $\cos(3x)/\cos(x)$ in terms of $\cos(x)$ and $\sin(x)$.

```
>> expand(cos(3*x)/cos(x));

                                      2
                             4 cos(x)  - 3
```

[New] **Problem 26:** linearize $\cos(3x)/\cos(x)$.

```
>> combine(expand(cos(3*x)/cos(x)),sincos);

                            2 cos(2 x) - 1
```

5 Determining Zero Equivalence

[New] **Problem 28:** simplify $\sqrt{997} - (997^3)^{1/6}$.

```
>> sqrt(997)-(997^3)^(1/6);

                                     1/6       1/2
                        - 991026973    + 997

>> simplify(%);

                                   0
```

[New] **Problem 29:** simplify $\sqrt{999983} - (999983^3)^{1/6}$.

```
>> sqrt(999983)-(999983^3)^(1/6);

                           1/2                         1/6
                    999983    - 999949000866995087

>> simplify(%);

                                   0
```

[New] **Problem 30:** recognize that $(2^{1/3} + 4^{1/3})^3 - 6(2^{1/3} + 4^{1/3}) - 6$ is 0.

```
>> simplify(expand((2^(1/3)+4^(1/3))^3-6*(2^(1/3)+4^(1/3))-6));

                                        0
```

[New] **Problem 31:** show that the function $\log(\tan(x/2 + \pi/4)) - \sinh^{-1}(\tan x)$ is 0 in $x = 0$ and that its derivative is 0.

```
>> f := ln(tan(1/2*x + PI/4)) - asinh(tan(x)):
>> eval(subs(f,x=0));

                                   0

>> combine(normal(combine(normal(expand(diff(f,x))),sincos)),sincos);

                                   0
```

6 The Complex Domain

Problem 35: compute the rectangular form $a + ib$ of $\log(3 + 4i)$.

```
>> rectform(ln(3+4*I));

                               ln(5) + I atan(4/3)
```

The real and imaginary part are obtained with `Re(..)` and `Im(..)` respectively.

[New] **Problem 36:** compute the rectangular form of $\tan(x + iy)$.

```
>> rectform(tan(x+I*y),{x,y});

                  sin(2 x)              I sinh(2 y)
            -------------------- + --------------------
            cos(2 x) + cosh(2 y)   cos(2 x) + cosh(2 y)
```

Here the set `{x,y}` given as second argument specifies that the variables x and y are real. By default all variables are considered as complex.

Problem 39: do not simplify $\sqrt{e^z} - e^{z/2}$ to 0 without any assumption on the imaginary part of z.

```
>> sqrt(exp(z))-exp(z/2);

                                   / z \         1/2
                             - exp | - | + exp(z)
                                   \ 2 /
```

7 Equations

[New] **Problem 47:** solve the polynomial equation $3x^3 - 18x^2 + 33x - 19 = 0$.

```
>> s:=solve(3*x^3-18*x^2+33*x-19):
>> simplify(rectform(s[i])) $ i=1..3;

   1/2     / PI \                       1/2     / PI \
 2 3    cos | -- |                      3    cos | -- |
            \ 18 /             / PI \            \ 18 /             / PI \
 ----------------- + 2, - sin | -- | - --------------- + 2 , sin | -- | -
         3                    \ 18 /          3                  \ 18 /
```

```
 1/2      / PI \
3    cos | -- |
          \ 18 /
--------------- + 2
       3
```

[New] **Problem 60:** solve the linear (dependent) system $x+y+z=6, 2x+y+2z=10, x+3y+z=10$.

```
>> linsolve({x+y+z=6,2*x+y+2*z=10,x+3*y+z=10},{x,y,z});

                        {y = 2, x = - z + 4 }
```

[New] **Problem 61:** solve the system of nonlinear equations $x^2y+3yz-4=0, -3x^2z+2y^2+1=0, 2yz^2-z^2-1=0$.

```
>> s:=solve({ x^2*y + 3*y*z - 4, -3*x^2*z + 2*y^2 + 1, 2*y*z^2 - z^2 - 1}):
>> subs(s, RootOf(18*z^8-48*z^7+21*z^6+5*z^4+3*z^2+1,z)=R);

 {
 {           /               2       3        4       5        6        7      2
 { x = RootOf \ 19 R - 48 R  + 2 R  + 48 R  + 3 R  - 48 R  + 18 R  + 6 x ,
 {

                          2       4                              }
              \         5 R    21 R          5       6           }
             x /, y = - ---- - ----- + 24 R  - 9 R  - 1 , z = R }
                         2       2                               }
```

Here the solution is expressed with the help of the `RootOf` notation. There are 8 solutions, each one corresponding to z being one of the roots of the polynomial $18z^8-48z^7+21z^6+5z^4+3z^2+1$, y is a polynomial in z as shown above, and x being one root of a degree two polynomial dependent on z.

8 Matrix Algebra

Problem 62: invert the 2×2 matrix $[[a,b],[1,ab]]$.

```
>> loadlib("domains"): export(domains):
>> loadlib("linalg"): export(linalg):
>> M := Matrix( ExpressionField(normal) ):
>> m := M( [[a,b],[1,a*b]] ):
>> 1/m;

+-                       -+
|    a           1        |
|  ------  ,  ---------   |
|   2            2        |
|  a  - 1     - a  + 1    |
|                         |
|    1           a        |
| --------  , ----------- |
|      2            2     |
| b - a  b   - b + a  b   |
+-                       -+
```

Here we first load the necessary packages, then we define the domain `M` of matrices whose coefficients are in the domain `ExpressionField(normal)`, which means the set of all MuPAD expressions after simplification by **normal**. We could have written as well **Matrix(ExpressionField)** if we do not want the expressions to be normalized. Please note that no special command is necessary to invert the matrix: we can use the standard arithmetic operators due to the overloading mechanism of MuPAD.

Problem 63: compute the determinant of the 4×4 Vandermonde matrix over w, x, y, z.

```
>> loadlib("linalg"): export(linalg):
>> M := SquareMatrix(4):
>> m := M([[1,1,1,1],[w,x,y,z],[w^2,x^2,y^2,z^2],[w^3,x^3,y^3,z^3]]):
>> Factor(det(m));

                (w - x) (w - z) (x - z) (x - y) (w - y) (y - z)
```

[New] **Problem 64:** find the eigenvalues of the matrix $[[5,-3,-7],[-2,1,2],[2,-3,-4]]$.

```
>> m := SquareMatrix(3)([[5, -3, -7],[-2,  1,  2],[ 2, -3, -4]]):
>> Id := SquareMatrix(3)(1,Diagonal):
>> det(m-lambda*Id);

    - 14 lambda + (- lambda + 1 ) (- lambda - 4 ) (- lambda + 5 ) + 14

>> solve(%,lambda);

                                 1, 3, - 2
```

Here we supposed that the **linalg** package was already loaded (see above example). It was already possible in MuPAD 1.2.1 to compute the determinant, but MuPAD 1.2.2 can now solve the characteristic polynomial too.

9 Sums and products

[New] **Problem 67:** find a closed-form for $\sum_{k=1}^{n} k^3$.

```
>> Factor(sum(k^3,k=1..n));

                                      2         2
                                 1/4 n   (n + 1)
```

[New] **Problem 68:** find a closed-form for $\sum_{k=1}^{\infty}(1/k^2 + 1/k^3)$.

```
>> sum(1/k^2+1/k^3,k=1..infinity);

                                           2
                                         PI
                                zeta(3) + ---
                                           6
```

10 Calculus

[New] **Problem 71:** compute the limit of $(1 + 1/n)^n$ when n goes to infinity, and the limit of $(1 - \cos x)/x^2$ when x goes to zero.

```
>> limit((1+1/n)^n,n=infinity);

                                     E

>> limit((1-cos(x))/x^2,x=0);

                                    1/2
```

It should be noted that the limit functions of MuPAD implements the best known algorithm to this date. It is based on the concept of most-rapidly-varying subexpressions (MRV), and a comparison made by Dominik Gruntz on 19 exp-log expressions [3] showed that the algorithms implemented in other computer algebra systems (except Maple and MuPAD) fail on many examples, and even return wrong values !

11 Indefinite Integrals

[New] **Problem 73:** integrate $1/(x^3+2)$, then differentiate and simplify.

```
>> int(1/(x^3+2),x);

           /                        /      3        \ \
       sum \ X2 ln(x + 6 X2), X2 = RootOf \ 108 X3  - 1, X3 / /

>> diff(%,x);

                                     1
                                   ------
                                    3
                                   x  + 2
```

The output from `int` means that an antiderivative of $1/(x^3+2)$ is the sum of the three functions $\alpha_i \ln(x+6\alpha_i)$, where α_i is one of the roots of the polynomial $108X^3-1$. The `sum` expression can be differentiated, and the result is a sum of three rational functions, namely $\alpha_i/(x+6\alpha_i)$, which is automatically normalized. More generally, the sum of any rational function over the roots of a polynomial simplifies to an expression involving the polynomial coefficients. The MuPAD syntax for this is `sum(f,t=RootOf(..))`; for example `sum(t^2,t=RootOf(x^3+a*x^2+b*x+c,x))` returns `a^2-2*b`.

[New] **Problem 76:** compute the derivative of $|x|$.

```
>> diff(abs(x),x);

                                  sign(x)
```

The extension of the functionality of MuPAD by such rules is very easy. For example to “teach” to MuPAD that the derivative of $\sin(f)$ with respect to x is $\cos(f)f'$, we simply write:

```
>> sin := funcattr(sin,"diff",proc(f,x) begin cos(f)*diff(f,x) end_proc);
```

[New] **Problem 77:** compute an antiderivative of $|x|$.

```
>> int(abs(x),x);

                                  x abs(x)
                                  --------
                                     2
```

Problem 80: compute an antiderivative of $x/(\sqrt{1+x}+\sqrt{1-x})$.

```
>> int(x/(sqrt(1 + x) + sqrt(1 - x)), x);

          / ln(x + 1) \ / x       \       / ln(- x + 1 ) \ /   x        \
      exp | --------- | | - + 1/3 | + exp | ------------ | | - - + 1/3  |
          \     2     / \ 3       /       \      2       / \   3        /
```

The result can be simplified using the `expand` command.
Problem 81: compute an antiderivative of $(\sqrt{1+x}-\sqrt{1-x})/2$, which is the same as above.

```
>> int((sqrt(1 + x) - sqrt(1 - x))/2, x);

          / ln(x + 1) \ / x       \       / ln(- x + 1 ) \ /   x        \
      exp | --------- | | - + 1/3 | + exp | ------------ | | - - + 1/3  |
          \     2     / \ 3       /       \      2       / \   3        /
```

MuPAD 1.2.1 was already able to solve these two examples, but you had to explicitly load the `intlib` package and to call `intlib::int`. Now there is a "toplevel" function `int` for definite and indefinite integration.

12 Definite Integrals

[New] **Problem 90:** compute the integral $\int_{x=0}^{a}\int_{y=0}^{b(1-x/a)}\int_{z=0}^{c(1-x/a-y/b)} 1\,dz\,dy\,dz$.

```
>> int(int(int(1,z=0..c*(1-x/a-y/b)),y=0..b*(1-x/a)),x=0..a);

                                 a b c
                                 -----
                                   6
```

13 Series

Problem 91: compute the first terms of the Taylor expansion of $1/\sqrt{1-v^2/c^2}$ at $v=0$.

```
>> series(1/sqrt(1-(v/c)^2),v=0);

                     - 2  2      - 4  4
                    c    v    3 c    v         /  6 \
                    ------- + --------- + 1 + O \ v  /
                       2          8
```

[New] **Problem 92:** compute the inverse of the square of the above expansion.

```
>> 1/%^2;

                           - 2  2          /  6 \
                        - c    v  + 1 + O \ v  /
```

This is possible in MuPAD because of the domain facility, and the fact that standard arithmetic operations can be overloaded for domain elements. This allows to get $O(x)$ as the result of $O(x) - O(x)$, whereas for example Maple gives 0.
[New] **Problem 93:** Compute the Taylor expansion of $\tan(x)$ at $x = 0$ by dividing the expansion of $\sin(x)$ by that of $\cos(x)$.

```
>> series(sin(x),x)/series(cos(x),x)=series(tan(x),x);

                 3       5                           3       5
                x     2 x       /  6 \              x     2 x       /  6 \
           x + -- + ---- + O \ x   / = x + -- + ---- + O \ x   /
                3      15                           3      15

>> bool(%);

                                  TRUE
```

Here again, the results of a `series` command can be combined with any arithmetic operation.

[New] **Problem 95:** compute the expansion of $\log\sinh(z) + \log\cosh(z+w)$ at $z = 0$.

```
>> series(ln(sinh(z))+ln(cosh(z+w)),z,3);

                                 z sinh(w)      /  2 \
           ln(z) + ln(cosh(w)) + --------- + O \ z   /
                                  cosh(w)
```

[New] **Problem 96:** subtract from the above expansion that of $\log(\sinh(z)\cosh(z+w))$.

```
>> %-series(ln(sinh(z)*cosh(z+w)),z,3);

                                  /  2 \
                               O \ z   /
```

[New] **Problem 97:** compute the expansion of $\log(\sin x/x)$ at $x = 0$.

```
>> series(ln(sin(x)/x),x);

                             2    4
                            x    x       /  5 \
                          - -- - --- + O \ x   /
                            6    180
```

[New] **Problem 100:** revert the Taylor expansion of $\sin y + \cos y$ at $y = 0$.

```
>> s:=series(sin(y)+cos(y),y=0);

                          2    3    4     5
                         y    y    y     y            /  6 \
                     y - -- - -- + -- + --- + 1 + O \ y   /
                         2    6    24   120

>> t:=revert(s);

               2              3                          5
        (y - 1)     2 (y - 1)              4   17 (y - 1)           /        6 \
    y + -------- + ---------- + (y - 1)  + ----------- - 1 + O \ (y - 1)   /
           2             3                          10
```

We can check the result by asking for `s @ t` that computes the composition of `s` and `t`.

14 Differential Equations

[New] **Problem 105:** solve the differential equation $f''(t)+4f(t) = \sin(2t)$ with the initial conditions $f(0) = f'(0) = 0$.

```
>> loadlib("ode"):
>> ode({diff(f(t),t,t)+4*f(t)=sin(2*t),f(0)=0,D(f)(0)=0},f(t));

 ode({D(f)(0) = 0, f(0) = 0, 4 f(t) + diff(f(t), t, t) = sin(2 t)}, f(t))

>> solve(%);

                                -- sin(2 t)   t cos(2 t) --
                                |  -------- - ----------  |
                                --    8           4      --
```

Here we first load the `ode` package, then we define the equation with the `ode` command, and simply call `solve` on this equation.

[New] **Problem 107:** solve the equation $x^2y'(x) + 3xy(x) = \sin x/x$.

```
>> solve(ode(x^2*diff(y(x),x)+3*x*y(x)=sin(x)/x,y(x)));

                                --      - 3     - 3        --
                                |  C3 x     - x    cos(x)  |
                                --                         --
```

[New] **Problem 108:** solve the equation $y'' + yy'^3 = 0$ for $y(x)$.

```
>> solve(ode(diff(y(x),x,x)+y(x)*diff(y(x),x)^3=0,y(x)));

                       --           /                        3    \ --
                       |  C5, RootOf \ 6 x + 6 C6 + 6 C4 y - y  , y /  |
                       --                                             --
```

This result says that the solution is either a constant function $y(x) = C_5$, or an algebraic function satisfying the equation $y(x)^3 = 6C_4y(x) + 6x + 6C_6$ where C_4 and C_6 are arbitrary constants.

[New] **Problem 109:** solve the equation $\frac{d}{dx}y(x,a) = ay(x,a)$.

```
>> solve(ode(diff(y(x,a),x)=a*y(x,a),y(x,a)));

                                      [C1 exp(a x)]
```

MuPAD recognizes equations with only one differential variable as ordinary differential equations.

[New] **Problem 111:** solve the system $x'(t) = x(t) - y(t), y'(t) = x(t) + y(t)$.

```
>> sys := {diff(x(t),t)-x(t)+y(t)=0,diff(y(t),t)-x(t)-y(t)=0}:
>> solve(ode(sys, {x(t),y(t)}));

 [x(t) = (I) C5 exp((1 + I) t) + (- I) C6 exp((1 - I) t), y(t) = C5 exp((

                        1 + I) t) + C6 exp((1 - I) t)]
```

[New] **Problem 112:** verify the above is a solution.

```
>> eval(subs([op(sys)],op(%)));

                                      [0 = 0, 0 = 0]
```

15 Operators

Problem 115: define the operator $L = (D-1)(D+2)$ where D is the differentiation operator.

```
>> L := (D-id) @ (D+2*id);

                         (- id + D )@(2 id + D)
```

In MuPAD, `id` stand for the identity, and `@` is the composition operator.

Problem 116: compute $L(f)$ where L is the above defined operator.

```
>> L(f);

                         - 2 f + D(f) + D(D(f))
```

Problem 117: compute $L(g)(y)$ where L is the above defined operator.

```
>> L(g)(y);

                     - 2 g(y) + D(g)(y) + D(D(g))(y)
```

Problem 119: define the operator T such that $T(f)(x,a) = f(a) + f'(x)(x-a) + f''(a)(x-a)^2/2$.

```
>> T:=proc(f) begin
&>    eval(subsop(hold(func(f,x,a)),
&>      1=_plus(f(a),_fconcat(D$k)(f)(a)/fact(k)*(x-a)^k$k=1..2)))
&> end_proc:
```

Problem 120: evaluate T for an unknown function f.

```
>> T(f);

      func(f(a) + (-a + x)*D(f)(a) + (-a + x)^2*D(D(f))(a)*1/2, x, a)
```

Problem 121: evaluate T for an unknown function g and a generic point (y, b).

```
>> T(g)(y,b);

                                                              2
                                         D(D(g))(b) (- b + y )
            g(b) + D(g)(b) (- b + y ) + ----------------------
                                                  2
```

Problem 122: evaluate T for the function sin and a point (z, c).

```
>> T(sin)(z,c);

                                                         2
                                            sin(c) (- c + z )
              sin(c) + cos(c) (- c + z ) - ------------------
                                                   2
```

16 Programming

Problem 123: compute the Legendre polynomials directly.

```
>> p:= proc(n, x) begin
    if n = 0 then 1
    else normal(1/(2^n*fact(n)) * diff((x^2 - 1)^n, x$n))
    end_if
    end_proc:
>> p(i,x) $ i=0..4;

                 2                    3         2       4
              3 x           3 x    5 x      15 x    35 x
        1, x, ---- - 1/2, - --- + ---- , - ----- + ----- + 3/8
               2             2      2         4       8
```

Problem 124: compute the Legendre polynomials recursively, using their recurrence of order 2.

```
>> pp:= proc(n, x) option remember; begin
     if n=0 then 1
     elif n=1 then x
     else normal(((2*n - 1)*x*pp(n - 1, x) - (n - 1)*pp(n - 2, x))/n)
     end_if
     end_proc:
>> pp(i,x) $ i=0..4;

                 2                    3         2       4
              3 x           3 x    5 x      15 x    35 x
        1, x, ---- - 1/2, - --- + ---- , - ----- + ----- + 3/8
               2             2      2         4       8
```

Problem 125: evaluate the fourth Legendre polynomial at 1.

```
>> pp(4,1);

                                  1
```

The Legendre polynomials are already defined in the `orthpoly` package, whence one can write `orthpoly::legendre(4,1)` too.

17 Translation

Problem 126: define the polynomial $p = \sum_{i=1}^{5} a_i x^i$.

```
>> p:= poly(_plus(a.i * x^i $ i=1..5), [x]);

               /          2       3       4       5         \
          poly \ x a1 + x  a2 + x  a3 + x  a4 + x  a5, [x] /
```

Problem 127: apply Horner's rule to the above polynomial.

```
>> p(x);

                x (a1 + x (a2 + x (a3 + x (a4 + x a5))))
```

18 Boolean Logic

Problem 129: evaluate `TRUE and FALSE`.

```
>> TRUE and FALSE;

                                FALSE
```

Problem 130: simplify `x or (not x)`.

```
>> x or (not x);

                                TRUE
```

Number of solved problems: 68 over 131 (39 new problems with respect to MuPAD 1.2.1). As a comparison, Reduce can solve 49 problems, Axiom 57 problems, Derive 67 problems, Mathematica 80 problems, Maple 89 problems and Macsyma 102 problems, but none of these systems is public domain ! You can get MuPAD by anonymous ftp from `ftp.uni-paderborn.de` (main site) or in France from `ftp.inria.fr`.

Acknowledgement. Many thanks to Michael Wester for his very detailed comments on a first version of this paper.

References

[1] FUCHSSTEINER, B., AND AL. *MuPAD Benutzerhandbuch.* Birkhäuser, Basel, 1993.

[2] FUCHSSTEINER, B., AND AL. *MuPAD Tutorial.* Birkhäuser, Basel, 1994.

[3] GRUNTZ, D. *On Computing Limits in a Symbolic Manipulation System.* PhD thesis, Swiss Federal Institute of Technology Zürich, 1995.

[4] WESTER, M. A review of cas mathematical capabilities. *Computer Algebra Nederland Nieuwsbrief 13* (Dec. 1994), 41–48.